

## Genetically Optimized Self-Organizing Polynomial Neural Networks

朴 浩 成\* · 朴 炳 俊\*\* · 張 成 煥\*\*\* · 吳 聖 權§

(Ho-Sung Park · Byoung-Jun Park · Sung-Whan Jang · Sung-Kwun Oh)

**Abstract** - In this paper, we propose a new architecture of Genetic Algorithms(GAs)-based Self-Organizing Polynomial Neural Networks(SOPNN), discuss a comprehensive design methodology and carry out a series of numeric experiments. The conventional SOPNN is based on the extended Group Method of Data Handling(GMDH) method and utilized the polynomial order (viz. linear, quadratic, and modified quadratic) as well as the number of node inputs fixed (selected in advance by designer) at Polynomial Neurons (or nodes) located in each layer through a growth process of the network. Moreover it does not guarantee that the SOPNN generated through learning has the optimal network architecture. But the proposed GA-based SOPNN enable the architecture to be a structurally more optimized network, and to be much more flexible and preferable neural network than the conventional SOPNN. In order to generate the structurally optimized SOPNN, GA-based design procedure at each stage (layer) of SOPNN leads to the selection of preferred nodes (or PNs) with optimal parameters- such as the number of input variables, input variables, and the order of the polynomial- available within SOPNN. An aggregate performance index with a weighting factor is proposed in order to achieve a sound balance between approximation and generalization (predictive) abilities of the model. A detailed design procedure is discussed in detail. To evaluate the performance of the GA-based SOPNN, the model is experimented with using two time series data (gas furnace and NOx emission process data of gas turbine power plant). A comparative analysis shows that the proposed GA-based SOPNN is model with higher accuracy as well as more superb predictive capability than other intelligent models presented previously.

**Key Words** : Self-Organizing Polynomial Neural Networks(SOPNN), Polynomial Neuron(PN), aggregate objective function, Genetic Algorithms(GAs), GMDH(Group Method of Data Handling), design procedure, time series data

### 1. 서 론

컴퓨터의 급속한 성장에 따라 첨단 인공지능 기법을 통한 모델 개발의 이론적 발전과 실무적용에서 상당한 성과를 얻고 있다. 특히, 모델의 이론적 개발은 입출력 데이터쌍의 집합으로부터 미지의 함수를 평가하는 것으로 이는 다양한 과학과 공학분야[1, 2]에서 여전히 중요한 관심이 된다. 시스템의 성능을 예측하기 위하여 시스템의 복잡성이 큰 문제를 분석하기 위한 모델링은 그 시스템의 변수들을 전문가의 시행착오에 의해서 구축되기 때문에 객관적인 모델을 구축할 수가 없었다. 또한 시스템이 복잡하고 대규모의 구조인 경우에는 함수식으로 모델의 표현이 한정되지 않기 때문에 회귀 분석을 적용할 수 없는 경우가 있다. 즉, 비선형 실 시스템의 동적 분석을 표현하는데 있어서의 수학적 모델들은 많은 입·출력 변수들 사이에서 모델을 구축하는 변수들과 모델 구조의 선택에 많은 문제를 가지고 있다. 이러한 문제를 풀기 위해, A. G. Ivakhnenko[3-6]는 GMDH(Group Method

of Data Handling) 알고리즘을 소개하였다. GMDH 형태 알고리즘은 복잡한 비선형 공정들의 예측과 모델링을 위해서 1970년대 중반이후부터 매우 광범위하게 사용되어져 왔다. GMDH의 주된 특성은 자기 구성과 적자생존의 자연 법칙에 근간을 둔 진화적 알고리즘이다. 이는 시스템의 입력과 출력사이의 비선형 관계를 동정하기 위한 분석적인 기술이다. GMDH를 기반으로 구축된 다항식 뉴럴 네트워크는 표준 인공 신경망보다 더 적은 수의 노드를 가지지만 적응성면에서는 더 우수한 특성을 가진다. 이러한 구조는 과학과 공학[7, 8]등 다양한 분야에서 사용되어 왔으며 우수한 예측 능력을 가진 모델을 구축한다. GMDH 알고리즘은 데이터의 부분 표현식의 사용에 의해서 각 층에서의 노드들과 층들이 자기 스스로 연속적인 생성을 통해 최적의 구조를 구축하는 알고리즘이다. 그러나 이는 2입력, 2차 회귀다항식의 형태로 복잡한 비선형 시스템을 표현하는데는 어느 정도 한계가 있다. 이러한 단점을 극복하고자 Oh et al. 등에 의해 자기구성 다항식 뉴럴 네트워크(Self-Organizing Polynomial Neural Networks : SOPNN)[9, 10]가 개발되었다. SOPNN의 구조는 전방향 뉴럴 네트워크의 구조와 비슷하며 단지 뉴럴 네트워크의 뉴런이 자기구성 다항식 노드들에 의해서 대체되는 구조일 뿐 뉴럴 네트워크와 같이 오류 역전파 알고리즘은 사용하지 않는다. 또한 GMDH의 2입력 2차 다항식에서 벗어나 다변수 다차식의 확장된 부분표현식을 이용하여 연속적인 층과 노드(Polynomial Neuron : PN)를 생성

\* 正 會 員 : 圓 光 大 學 制 御 計 測 工 學 科 博 士 課 程

\*\* 正 會 員 : 圓 光 大 學 電 氣 電 子 工 學 部 博 士

\*\*\* 正 會 員 : 圓 光 大 學 電 氣 電 子 工 學 部 教 授 · 工 博

§ 正 會 員 : 圓 光 大 學 電 氣 電 子 工 學 部 副 教 授 · 工 博

接 受 日 字 : 2003 年 5 月 2 日

最 終 完 了 : 2003 年 10 月 24 日

함으로써 최적의 구조를 얻는 알고리즘으로 기존의 신경망 알고리즘보다 더 적은 노드수와 유연한 적응특성을 가지고 있다. 이러한 SOPNN 성능은 기존의 퍼지 모델링 방법보다 우수한 성능을 보여준다[9, 10]. 그러나 SOPNN 구조의 성능을 향상시키는데 크게 영향을 미치는 노드의 입력변수의 수와 차수 그리고 입력변수는 설계자의 경험에 의해서 결정함으로써 최적의 네트워크를 구축하기 위해서는 다양한 네트워크 구조와 그와 관련된 다양한 방법론적 설계가 필요하였다. 즉 최적 모델 구축에 있어서 객관적이고 신뢰성이 떨어지는 단점이 있다. 따라서 경험자의 경험에서 벗어나 모델이 최적으로 구축되기 위한 노드의 입력변수의 수와 이에 해당되는 입력변수의 선택 그리고 차수의 선택이 필요하다.

본 논문에서 제안된 진화론적 최적 자기 구성 뉴럴 네트워크(Self-Organizing Neural Networks : SOPNN)는 최적화 탐색 방법인 유전자 알고리즘[11-14]을 이용하여 자기구성 다항식 뉴럴 네트워크의 입력변수의 수와 이에 해당되는 입력변수 그리고 다항식의 차수를 탐색하여 최적의 자기구성 다항식 뉴럴 네트워크를 구축한다. 따라서 모델 구축에 있어서 유연성과 정확성을 가지며 객관적이고 좀 더 정확한 예측 능력을 가진 SOPNN 모델을 구축할 수가 있다. 또한 기존의 SOPNN 모델에 있어서 노드의 성능평가는 단순히 학습 데이터 또는 테스트 데이터 둘 중 하나로 국한되어 있는 반면, 본 논문에서는 하중계수를 가진 목적함수[15]를 이용하여 모델의 근사화와 일반화 능력사이의 합리적 균형을 통하여 노드의 성능을 평가함으로써 상호 연계된 최적 모델은 구축할 수가 있다. 유전자 알고리즘은 모든 진화 기반의 탐색 알고리즘 가운데 유연하고 강인한 탐색능력으로 인해 불연속 함수 문제를 포함한 최적화문제를 해결할 수 있는 알고리즘으로 다른 진화 알고리즘과 달리 파라미터 그 자체(표현형)가 아니라 파라미터의 코딩(유전자형)을 가지고 동작한다면 점과 해 공간에서 한 점이 아니라 점의 집단에서 동작한다는 특징을 가지고 있다.

본 논문의 평가를 위해서 Box와 Jenkins가 사용한 가스 토. 입출력 데이터[16]와 가스 터빈 발전소의 NOx 배출 공장 데이터[17]를 이용하여 제안된 모델의 타당성, 정확성 및 예측 성능 등을 기존 지능 모델들과 비교를 통해 그 우수성을 보인다.

## 2. 자기구성 다항식 뉴럴 네트워크

기존의 SOPNN[9,10]은 층들의 구성과 노드들 사이의 상호 연결성뿐만 아니라 각각의 노드들의 유연성 때문에 매우 다양한 구조 생성이 가능하다. SOPNN이 뉴로-퍼지 구조와 구별되는 가장 중요한 점은 그 구조의 학습이 네트워크 구조상의 전개에 중요한 역할을 하는 확장된 GMDH 알고리즘을 기반으로 한다는 것이다. GMDH 알고리즘은 모델링하고자 하는 대상 시스템의 입력변수  $x_i$ 와  $x_j$ , 출력  $y$ 의 쌍에 대해서 2차 회귀방정식을 계산함으로써 이 알고리즘을 시작한다.

$$y = A + Bx_i + Cx_j + Dx_i^2 + Ex_j^2 + Fx_ix_j \quad (1)$$

식 (1)을 사용하여 기존의  $m$ 개의 입력  $x_1, x_2, \dots, x_m$ 에서 출력  $y$ 를 예측하는 2개의 입력을 선택하여  $m(m-1)/2$ 개의 변수 수 즉, 노드의 수를 얻는다. 이 새로운 입-출력 관계로

부터 2차 회귀다항식을 찾은 후 이 가운데에서  $y$ 를 가장 잘 예측하는 2차 회귀 다항식 구조만이 판단기준을 통하여 선택한 다음 선택된 2차 회귀 다항식 구조의 결과값들을 다음 층의 입력으로 대치한다. 이 각각의 2차 회귀다항식을 통한 결과값을 이용하여 새로운 독립적 관찰결과를 얻는다. 즉, 새로운 입력변수를 구성한다. 이 새로운 입력변수를 가지고 앞에 했던 것처럼 노드의 수 결정과 2차 회귀다항식을 구성한다. 이 2차 회귀다항식들이 이전의 것보다 예측 능력이 떨어지기 시작하면 이 반복 과정을 중지한다. 이런 일련의 과정이 끝난 후, 가장 마지막 단계에서 얻어진 다항식들 중에서 가장 좋은 이차 다항식을 선택한다. 이 선택된 다항식 변수에 전 층에서의 2차 회귀다항식을 대입하고, 동일한 조작을 첫 번째층까지 반복한다. 그런 다음 최종적으로 추정 모델을 얻는다. 다시 말해서, 대수적 대입 법칙을 사용해서 아래의 식과 같은 매우 복잡한 형태의 다항식을 얻을 수 있다.

$$\hat{y} = a + \sum_{i=1}^m b_i x_i + \sum_{i=1}^m \sum_{j=1}^m c_{ij} x_i x_j + \sum_{i=1}^m \sum_{j=1}^m \sum_{k=1}^m d_{ijk} x_i x_j x_k + \dots (2)$$

이 식을 Ivakhnenko 다항식이라 한다.

위에서 살펴본 GMDH 알고리즘을 기반으로 확장된 SOPNN 설계 알고리즘을 다음과 같이 단계별로 설명한다.

### [단계 1] 시스템 입력변수의 결정

출력변수  $y$ 에 관계하는  $n$ 개의 시스템 입력변수를 결정한다. 이 시스템 입력변수를  $x_1, x_2, \dots, x_n$ 이라 한다.

### [단계 2] 데이터 분할

$N$ 개의 입출력 데이터  $(x_i, y_i) = (x_{1i}, x_{2i}, \dots, x_{ni}, y_i), i=1, 2, \dots, N$ 을 학습용 데이터와 테스트용 데이터로 분할한다. 학습용 데이터는 모델을 동정하기 위해 사용하는 데이터이고, 테스트용 데이터는 모델을 테스트하기 위해 사용하는 데이터이다. 학습용 데이터와 테스트용 데이터의 개수를 각각  $N_L$ 개와  $N_C$ 개로 한다. 여기서,  $N=N_L+N_C$ 이다.

### [단계 3] SOPNN 구조 결정

기존의 SOPNN 알고리즘 구조는 그림 1에서 보여준 것과 같이 다양한 구조 선택을 통하여 주어진 시스템 환경의 변화에 적응할 수 있는 유연성을 가진 모델을 구축할 수가 있다.

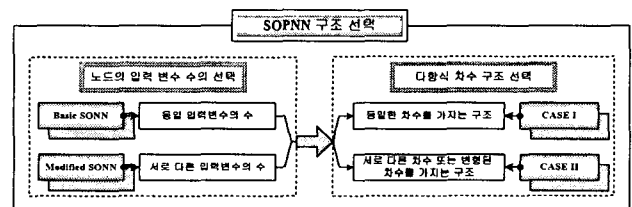


그림 1 SOPNN 구조에서 이용할 수 있는 설계  
Fig. 1 The design available in SOPNN architecture

각 층 노드의 다항식 차수 선택은 표 1에 표시한 것 같이, 동일 차수를 사용할 경우와 서로 다른 차수 또는 변형된 차수를 사용할 경우로 나눌 수 있다.

### [단계 4] 입력변수 수와 회귀 다항식 차수 선택

입력변수  $x_1, x_2, \dots, x_n$ 을 가지고 선택한 모델에 따라 1층과 2층 이상부터 최종 출력층까지 각 노드의 입력변수 수와 다항식 차수를 결정한다. 각 층에 사용될 다항식은 다음 표 2와 같다.

표 1 전체 SOPNN 구조

Table 1 A family of the topology of the SOPNN

층	입력변수 수	다항식 차수	SOPNN 구조
1층	p	Type P	(1) p=q : Basic SOPNN a) P=Q : CASE I b) P≠Q: CASE II
2층 이상	q	Type Q	(2) p≠q : Modified SOPNN a) P=Q : CASE I b) P≠Q: CASE II

(p, q=3, 4, 5), (P, Q=1, 2, 3)

표 2 회귀다항식의 서로 다른 형태

Table 2 Different forms of the regression polynomials standing

	1	2	3
1(Type 1)	Linear	Bilinear	Trilinear
2(Type 2)	Quadratic	Biquadratic	Triquadratic
2(Type 3)		Modified Biquadratic	Modified Triquadratic

선택된 입력의 개수에 따라 층 노드의 개수는  $n!/((n-r)!r!)$  과 같이 정의된다( $n$ :전체 입력의 수,  $r$ :선택한 입력의 수). 노드의 차수 및 입력변수 수 선택은 시스템의 특성, 모델 설계방법, 비선형성 및 예측 성능등을 고려함으로써 제기된 문제에 대해 가장 좋은 선택구조를 결정하는데 도움을 준다

[단계 5] 모델 테스트 및 노드의 선택

선형 회귀분석에는 학습 데이터를 사용하고, 기존의 연구 결과와 비교하기 위하여 성능지수 즉, 동정오차는 식 (3)과 같이 MSE(Mean Squared Error)을 이용한다.

$$E = \frac{1}{N_i} \sum_{m=1}^{N_i} (y_m - \hat{y}_m)^2 \quad (3)$$

여기서,  $N_i$ 는 학습데이터 총 수이다.

추정한 후반부 다항식에 테스트 데이터를 사용하여 식 (3)의 동정오차  $E$ 를 계산한다.

현재 층에서 얻어진  $n!/((n-r)!r!)$ 개의 동정오차들  $E_1, E_2, \dots, E_{n!/((n-r)!r!)}$ 에 대해 값이 작은순으로 배열한다.

1층에서 얻어진 노드 수는  $n!/((n-r)!r!)$ 개의 모두를 사용하며, 2층 이상부터는 선택한 임의의 노드 수  $W$ 로부터, i) 노드 수가  $n!/((n-r)!r!) > W$ 일 경우, 다음 층의 노드 수를  $W$ 개로 설정하여,  $E_1, E_2, \dots, E_W$ 와 같이 재배열하고 나머지  $n!/((n-r)!r!) - W$ 개의 모델은 버린다.

ii) 노드 수가  $n!/((n-r)!r!) \leq W$ 일 경우, 다음 층의 노드 수를  $n!/((n-r)!r!)$ 개로 설정하여  $E_1, E_2, \dots, E_{n!/((n-r)!r!)}$ 과 같이 재배열한다.

[단계 6] 종료판정

단계 5의 현재 층에서 얻어진 최소 동정오차  $E_1$ 이 다음의 부등식을 만족하는 경우에 알고리즘을 종료한다.

$$E_1 \geq E_s \quad (4)$$

단,  $E_s$ 는 이전 층의 최소 동정오차이다.

[단계 7] 다음 층의 새로운 입력변수 설정

현재 층에서 보존된 노드의 출력들( $z_{1i}, z_{2i}, \dots, z_{wi}$ )에서 다음 층의 새로운 입력들( $x_{1j}, x_{2j}, \dots, x_{wj}$ )로,  $x_{1j} = z_{1i}, x_{2j} = z_{2i}$ ,

$\dots, x_{wj} = z_{wi}$ 으로 구성되고, [단계 4]로 간다. 이 후, [단계 4]부터 [단계 7]까지를 반복한다. 알고리즘이 종료할 경우, 최소 동정오차  $E_s$ 을 얻은 노드의 다항식에 전 층의 출력을 대입하고 동일한 조작을 제 1층까지 반복하여 최종 추정 모델  $\hat{y}$ 를 얻는다.

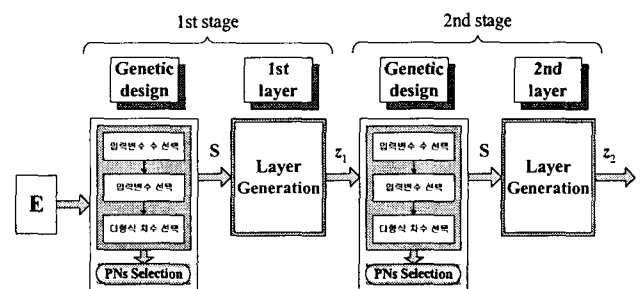
3. 자기구성 다항식 뉴럴 네트워크에서의 최적화

3.1 유전자 알고리즘

최근까지 여러 분야에서 일어나는 다양한 최적화 문제들을 해결하기 위해서 많은 기법들이 개발되고 문헌을 통해 발표되어 왔다[14]. 그 중에서 유전자 알고리즘(Genetic Algorithms : GAs)은 유전학과 자연 진화를 흉내낸 적응 탐색법으로 1975년 Holland[[11]에 의해 개발되었다. 유전자 알고리즘은 모든 진화 기반의 탐색 알고리즘 가운데 유연하고 강인한 탐색능력으로 인해 불연속 함수 문제를 포함한 최적화문제를 해결할 수 있는 알고리즘이다[12]. 유전자 알고리즘이 다른 진화 알고리즘과 다른 특성은 파라미터 그 자체(표현형)가 아니라 파라미터의 코딩(유전자형)을 가지고 동작한다면 점과 해 공간에서 한 점이 아니라 점의 집단에서 동작한다는 것이다.

본 논문에서는 유전자 알고리즘을 시뮬레이션하기 위해 설정된 각각의 개체에 존재하는 염색체의 수와 군집의 크기에 대한 파라미터의 정의로서, 한 개체에 여러 변수의 정보를 담은 직렬 방식을 사용한다. 재생산(Reproduction)연산으로 roulette-wheel 방식, 돌연변이연산은 선택된 비트를 반전시키는 invert 방식, 교차(crossover)연산은 one-crossover 방식을 사용하였다. 그러나 roulette-wheel 방식은 확률적 속성 때문에 선택 과정에서 최적자를 반드시 선택하지 못하는 단점을 가진다. 이를 보완해줄 수 있는 한 방법이 엘리트 전략(Elitist strategy)이다. 엘리트 전략은 만약 이전의 세대의 최적 개체를 저장하고 있다가 일련의 모의진화가 수행된 후 현 세대에서 소멸된 것이 확인되면, 저장된 최적 개체를 현 세대에서 가장 약한 개체 또는 임의로 선택된 개체와 교환하게 방법으로 roulette-wheel 방식의 단점을 극복하는 한 방법으로 유전자 알고리즘의 성능을 개선하는 것으로 알려졌다[13].

그림 2는 유전자 알고리즘을 통한 SOPNN 각 층의 최적화과정을 보여주고 있다.



E : Entire inputs, S : Selected PNs, z<sub>i</sub> : Preferred outputs in the i-th stage (z<sub>1</sub>, z<sub>2</sub>, ..., z<sub>n</sub>)

그림 2 유전자 알고리즘을 사용한 SOPNN 전체 최적화과정 Fig. 2 Overall optimization process of SOPNN using GAs

본 논문에서는 기존의 자기구성 다항식 뉴럴 네트워크가 가지고 있던 단점인 각 노드의 입력변수의 수와 이에 해당하는 입력변수 그리고 다항식 차수의 선택을 최적화 알고리즘인 유전자 알고리즘을 통해 탐색하여 경험자의 경험에 의존한 구조의 최적화를 객관적이고 효율적으로 구축하게 된다.

### 3.2 하중계수를 가진 목적함수

입출력 데이터로 시스템을 모델링 할 때 가장 문제가 되는 것은 구축된 모델이 학습 데이터만을 사용한 모델로써 시스템을 구축하는 경우이다. 이는 대상 공정의 모델을 주어진 데이터에 의해서만 성능향상을 꾀하였기 때문에 실 공정에 적용하면 많은 오차를 발생시키고 정확한 예측능력을 가지지 못한다. 이 문제를 해결하기 위하여 본 논문에서는 주어진 학습 데이터를 이용하여 자기구성 다항식 뉴럴 네트워크 모델을 구축하고 테스트 데이터를 사용하여 모델을 테스트하기 위하여 근사화 및 일반화의 상호 균형과 의존 능력을 가진 합성 목적 함수(성능지수)를 이용하여 상호 연계된 최적 모델 구조를 생성한다[15]. 이를 위해 학습 및 테스트 데이터의 성능 결과에 대한 하중계수를 가진 목적함수인 합성 성능지수로써 식 (5)와 같이 정의한다.

$$f(PI, EPI) = \theta \times PI + (1 - \theta) \times EPI \quad (5)$$

하중계수를 가진 목적함수인 식 (5)를 유전자 알고리즘과의 연계를 위해서 유전자 알고리즘의 적합도 함수로 다음과 같이 식 (6)을 사용한다.

$$F(\text{Fitness Function}) = \frac{1}{1 + f(PI, EPI)} \quad (6)$$

$\theta$ 는  $PI$ 와  $EPI$ 에 대한 하중계수를 나타내고,  $PI$ 는 학습 데이터에 대한 성능지수,  $EPI$ 는 테스트 데이터에 대한 성능지수를 나타낸다.

## 4. 자기구성 다항식 뉴럴 네트워크 모델의 진화론적 설계

본 장에서는 기존의 자기구성 다항식 뉴럴 네트워크를 유전자 알고리즘을 이용하여 자기구성 다항식 뉴럴 네트워크를 최적화하는 설계절차를 나타낸다. 기존의 자기구성 다항식 뉴럴 네트워크는 각 층의 입력변수의 수와 다항식 차수를 설계자가 경험에 의해서 결정함으로써 최적의 네트워크를 구하는데 있어서 다양한 네트워크 구조와 그와 관련된 다양한 방법론적 설계가 필요하였다. 이 다양한 구조에 최적화 방법인 유전자 알고리즘을 접목하여 단일화된 최적 네트워크 구조를 구축하고자 한다. 유전자 알고리즘을 이용한 자기구성 다항식 뉴럴 네트워크의 최적화 설계방법은 다음과 같다.

#### [단계 1] 시스템 입력변수의 결정

출력변수  $y$ 에 관계하는  $n$ 개의 시스템 입력변수를 결정한다. 필요하다면 데이터를 정규화한다. 이 시스템 입력변수를  $x_1, x_2, \dots, x_n$ 이라 한다.

#### [단계 2] 데이터 분할

$N$ 개의 입출력 데이터  $(x_i, y_i) = (x_{i1}, x_{i2}, \dots, x_{in}, y_i), i=1, 2, \dots, N$ 을 학습용 데이터와 테스트용 데이터로 분할한다. 학

습용 데이터는 모델을 동정하기 위해 사용하는 데이터이고, 테스트용 데이터는 모델을 테스트하기 위해 사용하는 데이터이다. 학습용 데이터와 테스트용 데이터의 개수를 각각  $N_L$ 개와  $N_C$ 개로 한다. 여기서,  $N = N_L + N_C$ 이다.

#### [단계 3] SOPNN 구조 정보 결정

SOPNN 구조를 구축하는 있어서 기본적인 정보를 결정한다. 즉, SOPNN의 a)종료조건 선택-최적 노드의 적합도 값 비교 또는 최대 층 수 결정, b)노드에 입력될 최대 입력변수의 수 결정, c)하나의 층에서 생성될 노드의 수 결정 그리고 d)목적함수의 하중계수 값을 결정한다.

#### [단계 4] 유전자 알고리즘을 이용한 SOPNN 구조 결정 및 입력변수 수와 회귀 다항식 차수 선택

유전자 알고리즘을 이용한 SOPNN 구조 결정은 먼저 그림 3에서 보여진 것과 같이 유전자 알고리즘에서 각각의 염색체에 의해서 구조 선택을 통한 주어진 시스템 환경의 변화에 적용할 수 있는 유연성을 가진 모델을 구축할 수가 있다. 그림 3은 SOPNN 구조의 각 층에서 각 노드에 해당되는 것이다.

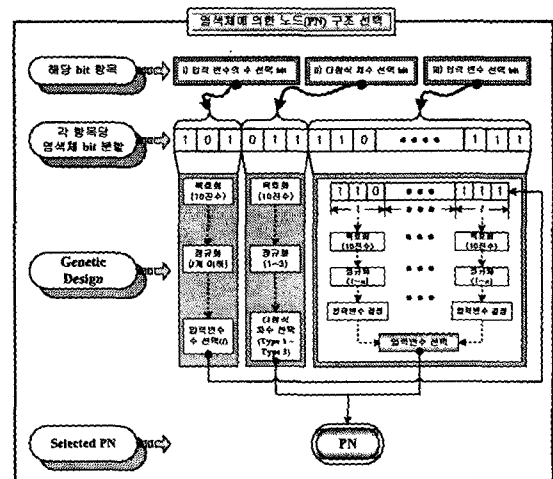


그림 3 유전자 알고리즘의 염색체를 이용한 SOPNN 구조에서 이용할 수 있는 PN 설계

Fig. 3 The PN design available in SOPNN architecture by using a chromosome of GAs

#### [단계 4-1] 입력변수의 수 선택

Step 1) 주어진 전체 염색체 중 처음 3비트를 입력변수의 수를 선택하는 비트로 설정한다.

Step 2) 랜덤하게 선택되어진 3비트를 식 (7)을 이용하여 10진수로 복호화 한다.

$$\beta = (2^2 \times bit(3)) + (2^1 \times bit(2)) + (2^0 \times bit(1)) \quad (7)$$

여기서,  $bit(1), bit(2), bit(3)$ 은 주어진 3비트의 위치를 나타낸다.

Step 3) 복호화된 10진수  $\beta$ 를 설계자가 정한 최대 입력변수의 수( $l$ )로 식 (8)을 이용하여 얻은 값을 반올림하여 정규화 한다.

$$\gamma = (\beta / \alpha) \times (l - 1) + 1 \quad (8)$$

여기서,  $\alpha$ 는 입력변수의 수 선택 비트에 해당되는 비트들이 모두 1로 설정된 경우에 10진수로 복호화한 값이다.

Step 4) 정규화된 값이 하나의 노드를 구축하는 입력변수의 수로 선택되어진다.

여기서, 입력변수의 수( $l$ )는 전체 입력변수인  $x_1, x_2, \dots, x_n$ 의  $n$ 보다 같거나 작은 수로 설정된다.

**[단계 4-2] 다항식 차수 선택**

Step 1) 입력변수의 수 선택 비트 다음의 3비트를 다항식 차수 선택을 위한 비트로 설정한다.

Step 2) 랜덤하게 선택되어진 3비트를 식 (7)을 이용하여 10진수로 복호화 한다.

Step 3) 복호화된 10진수를 식 (8)을 이용( $l$  대신 3을 대입)하여 1~3 사이의 값으로 정규화 한다.

1이 선택된 경우 -> Type 1이 사용 즉, 다항식이 선형식으로 구성된다.

2가 선택된 경우 -> Type 2이 사용 즉, 다항식이 2차 다항식으로 구성된다.

3이 선택된 경우 -> Type 3이 사용 즉, 다항식이 변형된 2차 다항식으로 구성된다.

각 Type별 다항식은 표 2에서 보여준다.

Step 4) 정규화된 값이 하나의 노드를 구축할 때 다항식 차수로 설정된다.

**[단계 4-3] 입력변수 선택**

Step 1) 위 두 부 염색체(Sub-chromosome) bit를 제외하고 나머지 비트들은 입력변수 선택을 위한 비트로 사용한다.

Step 2) 입력변수 선택을 위한 비트를 [단계 4-1]에서 구한 입력변수의 수로 균등 분할한다. 균등분할이 안될 때에는 다음과 같이 비트를 분할한다. 즉 입력변수 선택을 위한 비트가 22비트이고 입력변수의 수가 4라 하면, 22를 4로 나누어 반올림 처리를 한다. 그러면 첫 번째, 두 번째, 세 번째 입력변수 선택을 위한 비트는 각각 6비트씩 할당을 받고 나머지 비트가 마지막 네 번째 입력변수 선택을 위한 비트로 할당이 된다.

Step 3) 분할된 각 비트들을 식 (7)을 이용하여 10진수들로 각각 복호화 한다.

Step 4) 복호화된 각각의 10진수들을 식 (8)을 이용하여 전체 입력변수의 수로 정규화 한다.

Step 5) 정규화된 값들이 하나의 노드를 구성할 때 노드의 입력변수가 된다. 만약 선택된 입력변수가 중복된다면 중복된 입력변수를 하나의 입력변수로 처리한다.

기존의 SOPNN 구조[9, 10]에서는 하나의 층에서 모든 노드가 모두 똑같은 입력변수의 수와 다항식 차수를 가지고 모델 구축에 있어서 다양성과 데이터 특성에 맞는 모델을 구축하지 못하는 단점이 있었다. 즉, 하나의 층이 그림 1에서 보여준 것과 같이 "Basic SOPNN", "Modified SOPNN", "CASE I" 그리고 "CASE II"와 같이 구분되어 구축되어 있다. 이에 비해 본 논문에서 제안한 유전자 알고리즘을 이용한 SOPNN 구조는 위에서 살펴본 그림 3과 같은 염색체 처리를 SOPNN 전체 층의 모든 노드에 각각 실행함으로써 기존의 SOPNN 구조의 일률적인 입력변수의 수와 다항식 차수에서 벗어나 모델을 최적화하기 위한 입력변수의 수와 그에 해당하는 입력변수 그리고 다항식 차수가 유전자 알고리즘에 의해 모델이 최적화되기 위해 선택되어진다. 즉, 하나의 층에서 모델이 최적화되기 위해 "Basic SOPNN", "Modified SOPNN", "CASE I" 그리고 "CASE II"처럼 구분되어 구축

되지 않고 층의 노드들은 서로 다른 형태의 유연한 구조를 가지게 된다.

**[단계 5] 모델의 구축과 테스트 그리고 노드의 선택**

기존의 SOPNN에서는 모델의 구축은 앞에서 설명하였듯이 일반적인 설계방법이었으나 본 논문에서는 그림 3에 보여진 것과 같이 유전자 알고리즘을 이용하여 모델의 최적화에 맞게 각 노드들을 유연하게 구축할 수 있다.

유전자 알고리즘에 의한 SOPNN 구조의 전체 최적화 과정은 다음과 같은 절차로 수행된다.

Step 1) SOPNN 모델 구축을 위한 유전자의 세대, Population 개수, 돌연변이율, 교배율, 그리고 염색체의 길이와 같은 유전자의 정보를 설정한다.

Step 2) 첫 번째 세대에서 설정된 population 개수만큼 염색체에 의한 즉, 최적 다항식 구조 선택을 위해 노드(PN)을 구축한다.

여기서, 하나의 population은 SOPNN 구조에서 하나의 노드와 같은 역할을 하며, 각각의 population은 그림 3에서 나타난 것과 같이 진화론적 최적 설계에 의해 실행한다. 다항식의 매개변수는 최소 자승법을 이용하여 구한다.

Step 3) Population에 의해서 생성된 노드의 성능테스트를 위해서 본 논문에서는 모델의 근사화와 일반화 능력 사이의 합리적 균형을 얻기 위해 하중값을 가진 목적함수 식 (5)을 이용하여 노드를 테스트한다. 또한 식 (5)에 의해서 구해진 성능지수를 유전자 알고리즘의 적합도 함수로 계산하기 위해서 식 (6)과 같이 변환하여 유전자 알고리즘의 적합도 함수로 사용한다.

Step 4) 다음 세대의 생성을 위해서 구해진 population 정보와 적합도 값들을 이용하여 선택연산, 교배, 돌연변이를 실행한다.

Step 5) 구해진 적합도 값들( $F_1, F_2, \dots, F_z$ )을 선호되는 적합도 값 순(내림차순)으로 정렬한다.

Step 6) 정렬된 적합도 값들 중에서 중복된 적합도 값들은 제거하고 하나의 적합도 값으로 처리한다. 이는 모델의 각 층을 구축할 때 중복된 노드를 제거하기 위함이다.

Step 7) 중복된 적합도 값들을 하나로 처리한 후 설계자가 SOPNN 구조를 설계하기전 결정된 노드의 제한된 개수  $W$ 개만큼 선호되는 적합도 값들을 선택한다.

i) 계산된 적합도 함수의 노드 수가  $z \geq W$ 일 경우, 계산된 적합도 함수의 노드 수를  $W$ 개로 설정하고 나머지  $z - W$ 개의 노드는 버린다.

ii) 계산된 적합도 함수의 노드 수가  $z < W$ 일 경우, 적합도 함수의 노드 수를  $z$ 개로 설정한다.

Step 8) Step 7)에서 선택되어진 적합도 값들을 가진 노드들을 다시 초기 population 번호 순서에 따른 오름차순으로 재정렬한다. 이는 SOPNN 최종 모델을 구축할 때 관련 층 및 노드들을 체계적으로 구축하기 위함이다.

Step 9) 다음세대에서의 엘리트 전략을 위해서 선택된 노드들 중에서 최적의 적합도 값을 가지는 노드를 선택한다.

Step 10) Step 4)에서 얻어진 정보를 가지고 다음 세대의 population을 생성한 후 Step 2)에서 Step 9)까지 반복한다. 여기서 Step 3)에서 생성된 적합도 값들 중에서 가장 낮은 적합도 값을 가진 노드를 Step 9)에서 구한 이전 세대의 최적의 적합도 값을 가진 노드로 대체하는 엘리트 전략을 실행

행한다.

Step 11) 이전 세대에서 구해진 적합도 값들과 현재 세대에서 구해진 적합도 값들을 결합하여 다시 Step 5)부터 Step 8)까지 반복한다.

Step 12) 정해진 세대까지 Step 10)부터 Step 11)을 반복 실행한다.

위에서 살펴본 바와 같이, 마지막 세대까지 유전자 알고리즘이 실행되면 최적의 노드들이 생성되며 생성된 노드들은 SOPNN 모델을 구축하는 하나의 층을 이루게 된다.

[단계 6] 종료판정

단계 5의 현재 층에서 얻어진 최적 노드의 최대 적합도 값  $F_1$ 이 다음 부등식을 만족하는 경우에 알고리즘을 종료한다.

$$F_1 \leq F. \quad (9)$$

단,  $F_1$ 는 이전 층의 최적 노드인 최대 적합도 값이다.

그러나 위와 같은 종료판정은 네트워크 생성 중에 최대 층수가 과대하게 발생할 우려가 있으며 그럴 경우 프로그램 실행 시 많은 시간과 계산량을 요구하게 된다. 또한 현재 층의 동정오차가 이전 층의 동정오차보다 크더라도 그 다음 순층에서의 동정오차가 이전 층들의 동정오차보다 더 적은 값이 나올 수 있는데도 프로그램을 종료시키는 단점이 있다. 이와 같은 단점을 보완하기 위해 알고리즘의 판정을 뒤의 경우와 병행하여 전체 층의 수를 최대 5층으로 제한하여 네트워크의 복잡성과 우수성의 상호 균형을 고려하여 알고리즘을 종료한다.

[단계 7] 다음 층의 새로운 입력변수 설정

현재 층에서 보낸 노드의 출력들( $z_{1i}, z_{2i}, \dots, z_{wi}$ )에서 다음 층의 새로운 입력들( $x_{1j}, x_{2j}, \dots, x_{wj}$ )로,  $x_{1j}=z_{1i}, x_{2j}=z_{2i}, \dots, x_{wj}=z_{wi}$ 로 구성하고, [단계 4]로 간다(여기서,  $j=i+1$ ). 이 후, [단계 4]부터 [단계 7]까지를 반복한다.

알고리즘이 종료할 경우, 마지막 층의 최적 적합도 값  $F_1$ 은 얻은 노드의 다항식에 입력이 되는 전 층의 출력을 대입하고 동일한 조작을 제 1층까지 반복하여 최종 추정 모델을 얻는다.

5. 시뮬레이션 및 결과고찰

5.1 Gas furnace 공정

제안된 유전자 알고리즘을 이용한 자기구성 다항식 뉴럴 네트워크의 성능을 테스트하기 위해서 가스로 시계열 데이터를 이용하였다. Box와 Jenkins[16]에 의해 사용된 가스로 시계열 데이터는 입력력 데이터인 가스 흐름율( $u(t)$ )과 연소된 이산화탄소 농도( $y(t)$ )의 가스로 공정으로부터 추출된다.

입력( $u(t)$ )은 가스 흐름율이고 출력( $y(t)$ )은 이산화탄소 농도인 1입력 1출력의 가스로 시계열 데이터 296쌍을 시뮬레이션 하기 위해 전체 시스템 입력을  $u(t-3), u(t-2), u(t-1), y(t-3), y(t-2), y(t-1)$ 과 출력  $y(t)$ 의 6입력 1출력으로 설정하여 사용한다. 또한 전체 시스템 입력 데이터 296개 중 처음 148개의 데이터는 학습 데이터로 이용하고 나머지 148개의 데이터는 테스트 데이터로 나누어 자기구성 다항식 뉴럴 네트워크를 모델링하고 유전자 알고리즘을 사용하여 모델을 최적화한다. 표 3은 유전자 알고리즘을 기반으로 자기구성 다항식 뉴럴 네트워크의 다항식을 구성하기 위한 유전자 알

고리즘의 염색체 정보와 자기구성 다항식 뉴럴 네트워크 구축을 위한 정보를 나타낸 것이다.

표 3 유전자 알고리즘과 자기구성 다항식 뉴럴 네트워크 정보  
Table 3 Computational overhead and a list of parameters of the GAs and the SOPNN

	매개변수	1층~5층
GA	Maximum generation	100
	Total population size	60
	Selected population size	30
	Crossover rate	0.65
	Mutation rate	0.1
	String length	3+3+30
SOPNN	No. of inputs to be selected( $l$ )	$1 \leq l \leq 5$
	Type(T)	$1 \leq T \leq 3$
	Weighting factor( $\theta$ )	$0 \leq \theta \leq 1$

$l, T$  : integer

표 3에서 string length가 3+3+30인 것은 처음 3비트는 입력변수의 수를 선택하기 위한 비트이고, 두 번째 3비트는 다항식 차수를 선택하기 위한 비트이고, 마지막 30비트는 입력변수를 선택하기 위한 비트를 나타낸 것이다.

그림 4는 하중계수의 변화에 따른 SOPNN 각 층의 성능지수를 나타낸 것이고, 그림 5는 층의 증가에 따른 SOPNN의 성능지수를 나타낸 것이다.

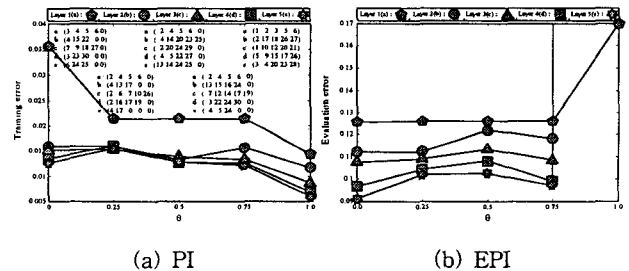


그림 4 하중계수의 변화에 따른 SOPNN의 성능지수  
Fig. 4 Performance index of SOPNN treated as a function of the weighting factor

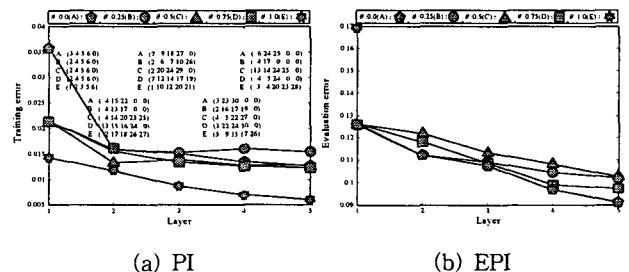


그림 5 층의 증가에 따른 SOPNN의 성능지수  
Fig. 5 Performance index of SOPNN according to the increase of layer

그림 4와 그림 5에서 보여지듯이 모델의 근사화와 일반화 사이의 상호 능력을 고려한다면  $\theta=0.75$ 인 경우 3층에서  $PI=0.013, EPI=0.108$ , 5층에서는  $PI=0.012, EPI=0.097$ 로 다른 하중계수일 때 보다 성능이 우수함을 알 수 있다. 그림 6은

$\theta=0.75$ 인 경우에 3층에서의 최적의 네트워크를 나타낸 것이다. 그림 4와 그림 5에서,  $a(\cdot)$ - $e(\cdot)$ ,  $A(\cdot)$ - $E(\cdot)$ 는 네트워크의 각 층에서의 최적의 노드 번호들을 나타내고 해당 층에서 최적의 네트워크 출력성능을 가진다. 즉, 그림 5에서 1층에서의 노드 번호는 시스템 입력번호를 나타내고, 2층 이상에서의 노드번호는 현재 층에서 가장 우수한 성능지수를 가지는 최적노드로서 앞선 층의 출력 노드번호를 나타낸다.

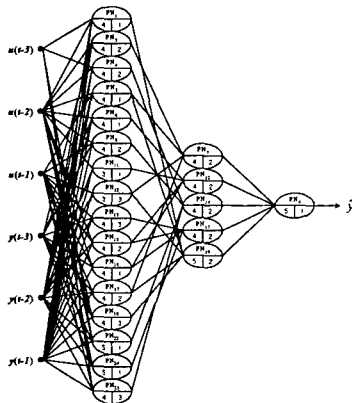
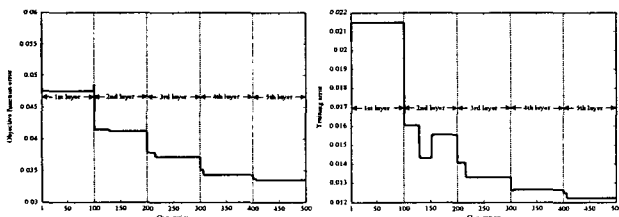


그림 6 진화론적 최적 자기구성 다항식 뉴럴 네트워크 구조 ( $\theta=0.75$ )

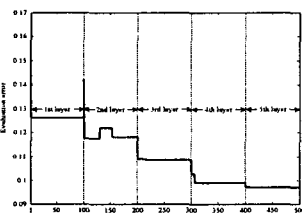
Fig. 6 Genetically optimized SOPNN architecture( $\theta=0.75$ )

그림 6에서 보여준 노드는 다음과 같이 표현된다. “ $\frac{PN_i}{NI/T}$ ” 여기서, ‘PN<sub>n</sub>’는 각 층에서 n번째 PN 노드를 나타낸 것이고, ‘NI’는 해당 노드로 들어오는 입력의 수를, ‘T’는 해당 노드에 사용된 차수를 나타낸 것이다. 그림 6에서 알 수 있듯이 기존의 SOPNN 구조와는 달리 각각의 층을 구성하는 노드들이 기존의 SOPNN 구조처럼 미리 설계자에 의해서 설정된 네트워크 구축이 아니라 GA의 구조적 동정에 의해 입력변수 수, 입력변수의 선택, 그리고 다항식 차수의 각 파라미터 값을 최적으로 선택하여 네트워크가 구축된다.



(a) f(PI, EPI)

(b) PI



(c) EPI

그림 7 유전자 알고리즘에 의한 각 성능지수들의 최적화 과정  
Fig. 7 The optimization process of each performance index by the genetic algorithms

그림 7은  $\theta=0.75$ 인 경우, 네트워크 각 층에서 generation 수행 과정에 따른 합성 성능지수, Training 성능지수 그리고 Testing 성능지수의 최적화 과정을 나타낸 것이다. 그림 7에서 알 수 있듯이 층이 증가할수록 성능은 개선되지만 계속 층을 증가하면 처리 속도와 계산량 증가와 같은 문제를 발생한다. 특히 4, 5층 부근에서 오차범위가 줄어드는 것을 알 수 있고, 정해진 오차의 tolerance level에서 최적의 네트워크를 구축 할 수 있음을 알 수 있다. 그러므로 SOPNN의 노드의 수를 줄이고, 층수 증가에 따른 iteration 시간을 줄이기 위해 그림 6에서 보여진 것 같이, 최대 3층까지의 최적 네트워크를 구성할 수 있다.

표 4는 가스로 공정 데이터를 사용하여 성능 지수를 식 (3)에 의해 계산된 값으로 기존의 모델과 본 논문에서 제시한 유전자 알고리즘을 이용한 SOPNN 모델과의 성능비교를 한 것이다. 표로부터 제안된 모델은 기존의 모델들과 비교해서 구조적으로 최적화된 유연한 자기구성 다항식 네트워크를 설계함으로써 우수한 성능을 나타냄을 보인다.

표 4 다른 모델링 방법과 성능 비교

Table 4 Comparison of performance with other modeling methods

Model		Performance Index		
		PI	PI <sub>s</sub>	EPI <sub>s</sub>
Oh and Pedrycz's model[15]		0.123	0.020	0.271
Tong's model[18]		0.469		
Sugeno and Yasukawa's model[19]		0.190		
Xu's model[20]		0.328		
Pedrycz's model[21]		0.320		
Kim, et al.'s model[22]			0.034	0.244
Lin and Cunningham's model[23]			0.071	0.261
Generic SOPNN[9]	Basic SOPNN	0.027	0.021	0.085
	Modified SOPNN	0.035	0.017	0.095
Advanced SOPNN[10]	Basic SOPNN		0.020	0.119
	Modified SOPNN		0.018	0.118
GA-SOPNN (Our model)	$\theta=0.0$		0.012	0.091
	$\theta=0.75$	0.022	0.012	0.097

표 4에서 “PI”는 전체 데이터를 이용하여 모델의 성능지수를 구한 것이고, “PI<sub>s</sub>”, “EPI<sub>s</sub>”는 전체 데이터를 학습 데이터와 테스트 데이터로 나누어 모델의 성능지수를 각각 구한 것이다.

### 5.2 가스 터빈 발전소의 NOx 배출 공정 데이터

대기상 질소산화물(NOx)은 주로 연소공정을 통해 배출되며 산업공정, 발전소, 자동차 배기가스 그리고 난방에 의해서 발생하는데, 그 중 대기에서 문제가 될 만큼 존재하는 것들은 N<sub>2</sub>O, NO 및 NO<sub>2</sub>이고 이들 물질은 대기오염 측면에서 NOx(질소 산화물)로 일컫는다.

NOx의 저감방법으로는 발생원으로부터 NOx의 배출을 억제하는 방법과 배출가스중의 NOx를 제거하는 방법을 들 수 있는데, 최근에는 Gas Turbine에 사용하는 입출력 변수 중에 입력으로 Tamb(The ambient temperature at site, in

degrees F), COM(The compressor speed, in rpm), LPT(The low pressure turbine speed, in rpm), Pcd(The compressor discharge pressure, in psia), Texh(The turbine exhaust temperature, in degrees F)등의 주변 환경변수를 조절하며 출력으로는 Parts per million-volume dry 단위의 NOx양으로 배출 메카니즘을 모델링하고, 최적 모델을 구축하여 시뮬레이터를 구성함으로써 NOx를 효과적으로 관리하는 방법이 연구되고 있다[17].

본 논문에서는 이러한 입출력 변수들을 이용하여 제안한 모델의 성능을 테스트하기 위해서 전체 260개의 데이터를 랜덤하게 130개씩 선택하여 학습데이터와 테스트 데이터로 사용하여 모델의 근사화와 일반화를 얻는다.

NOx 배출 공정 데이터를 유전자 알고리즘을 기반으로 자기구성 다항식 뉴럴 네트워크의 모델을 구축하기 위해 유전자 알고리즘의 탐색체 정보와 자기구성 다항식 뉴럴 네트워크 구축을 위한 정보는 표 3과 같다.

그림 8은 하중계수의 변화에 따른 SOPNN 각 층의 성능지수를, 그림 9는 층의 증가에 따른 SOPNN의 성능지수를 나타낸 것이다.

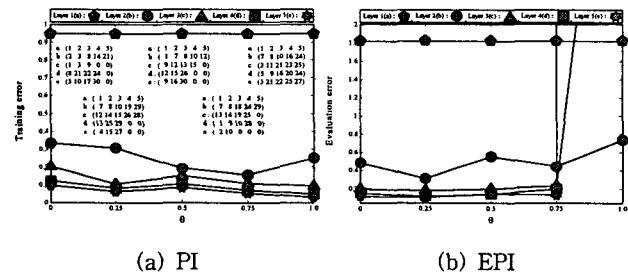


그림 8 하중계수의 변화에 따른 SOPNN의 성능지수  
Fig. 8 Performance index of SOPNN treated as a function of the weighting factor

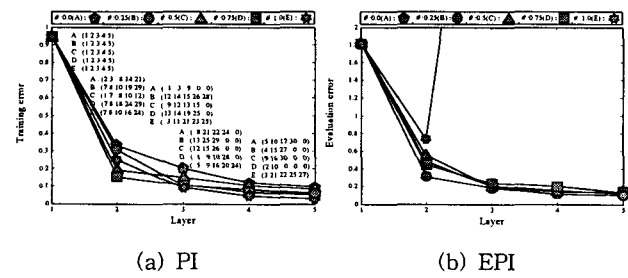


그림 9 층의 증가에 따른 SOPNN의 성능지수  
Fig. 9 Performance index of SOPNN according to the increase of layer

그림 8과 그림 9에서 보여지듯이, 1층인 경우엔 하중계수를 가진 목적함수의 하중계수 값에 상관없이 모두 똑같은 성능지수 값이 나옴을 알 수 있는데 이는 전체 입력이 5입력이고 또한 선택된 입력이 5입력 이하로 선택했기 때문으로 다항식 차수가 2차식인 경우 최적의 모델을 구축함을 보여준다. 가스로 공정 데이터와 달리 NOx 배출 공정 데이터와 같이 비선형성이 강하고 데이터가 분산된 구조일수록 시뮬레이션 결과에서 보여진바와 같이 본 논문에서 제안한 방법

이 성능을 개선시키고 최적의 모델을 구축하는데 유효함을 알 수 있다. NOx 배출 공정 데이터를 사용하여 모델을 구축한 결과 하중계수 값이  $\theta=0.25$ 인 경우에  $PI=0.062$ ,  $E\_PI=0.107$ 인 값을 얻을 수 있다.

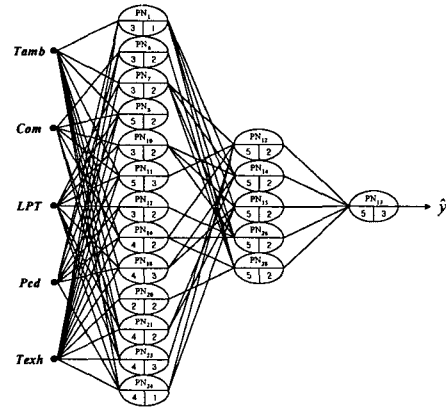


그림 10 진화론적 최적 자기구성 다항식 뉴럴 네트워크 구조 ( $\theta=0.25$ )

Fig. 10 Genetically optimized SOPNN architecture ( $\theta=0.25$ )

그림 10은 하중계수 값이  $\theta=0.25$ 인 경우에 3층에서 네트워크를 나타낸 것이다.

표 5는 가스 터빈 발전소의 NOx 배출 공정 데이터를 사용하여 성능 지수를 식 (3)에 의해 계산된 값으로 기존의 모델과 본 논문에서 제시한 유전자 알고리즘을 이용한 SOPNN 모델과의 성능비교를 한 것이다.

표 5 기존 모델방법과의 동정 오차비교  
Table 5 Comparison of performance with other modeling methods

Model	Performance Index		
	PI	PI <sub>s</sub>	EPI <sub>s</sub>
Regression model			
		17.68	19.23
Ahn's model[24]	FNN	5.835	
	AIM	8.420	
FNN[25] (Gas+Complex)	Simplified $\theta=0.4$	6.269	8.778
	Linear $\theta=0.2$	3.725	5.291
Multi-FNN[26]	Linear $\theta=0.75$	0.720	2.025
Our model	$\theta=0.25$	0.060	0.062

6. 결 론

본 논문에서는 최적화 방법인 유전자 알고리즘을 이용하여 진화론적 최적 자기구성 다항식 뉴럴 네트워크의 효율적인 설계 방법을 제안하였고 이를 기반으로 주어진 비선형 예제들을 통하여 GA기반 SOPNN의 우수성을 입증하였다.

기존의 SOPNN 알고리즘은 설계자가 모델을 구축하기 전 SOPNN 알고리즘 성능을 향상시키는데 크게 영향을 미치는 층을 구성하는 노드의 입력변수의 수와 다항식 차수 그리고 고정된 입력변수 선택을 설계자의 경험에 의해서 결



정함으로써 객관적이고 주어진 데이터의 특성에 맞는 최적의 모델을 구축하지 못하였고, 또한 최적의 모델을 구축하기 위해서 많은 시행착오를 필요로 하였다. 이러한 SOPNN 모델의 구조의 한계를 극복하고자 본 연구에서는 최적화 방법인 유전차 알고리즘을 이용하여 노드의 입력변수의 수와 그에 해당하는 입력변수 그리고 다항식 차수 등의 파라미터를 최적 선택하여 모델 구축에 있어서 유연성과 정확성을 가지며 더욱 정확한 예측 능력을 가진 최적화된 SOPNN 모델 구조를 구축할 수가 있었다. 또한 기존의 SOPNN 모델에 있어서 노드의 성능평가는 단순히 학습 데이터 또는 테스트 데이터 둘 중 하나로 국한되어 있는 반면, 본 논문에서는 하중계수를 가진 목적함수를 이용하여 모델의 근사화와 일반화 능력사이의 상호 균형을 통하여 노드의 성능을 평가함으로써 overfitting과 편중된 최적화(높은 근사화와 낮은 예측 성능으로 이끄는)를 완화하는 유연성을 가진 최적 모델을 구축할 수가 있었다.

각 공정의 시뮬레이션을 통해서 알 수 있듯이 제안한 모델은 유전차 알고리즘을 이용한 각 노드들의 유기적인 조직으로부터 기존의 다른 모델들보다 성능이 우수함을 알 수 있으며, 데이터 특성이 비선형성이 강하거나 또는 제한된 데이터의 수 일지라도 구조적 및 파라미터 동정에 의한 효율적인 최적 자기구성 다항식 뉴럴 네트워크를 구축할 수 있음을 보였다.

**감사의 글**

본 논문은 2003년도 원광대학교의 교비지원에 의해서 연구됨.

**참 고 문 헌**

[1] V. Cherkassky, D. Gehring, and F. Mulier, "Comparison of adaptive methods for function estimation from samples", *IEEE Trans. Neural Networks*, vol. 7, pp. 969-984, July 1996.

[2] J. A. Dicherson and B. Kosko, "Fuzzy function approximation with ellipsoidal rules", *IEEE Trans. Syst., Man, Cybern. Part B*, vol. 26, pp.542-560, Aug. 1996.

[3] A. G. Ivakhnenko, "Polynomial theory of complex systems", *IEEE Trans. on Systems, Man and Cybernetics*, Vol. SMC-1, pp. 364-378, 1971.

[4] A. G. Ivakhnenko and H. R. Madala, *Inductive Learning Algorithms for Complex Systems Modeling*, CRC Press, London, 1994.

[5] A. G. Ivakhnenko and G. A. Ivakhnenko, "The Review of Problems Solvable by Algorithms of the Group Method of Data Handling(GMDH)", *Pattern Recognition and Image Analysis*, Vol. 5, No. 4, pp. 527-535, 1995.

[6] A. G. Ivakhnenko, G. A. Ivakhnenko and J.A. Muller, "Self-organization of Neural Networks with Active Neurons", *Pattern Recognition and Image Analysis*, Vol. 4, No. 2, pp. 185-196, 1994.

[7] V. Sommer, P. Tobias, D. Kohl, H. Sundgren, and L. Lundstrom, "Neural networks and abductive networks for chemical sensor signals: A case comparison", *Sensors and Actuators, B*, 28, pp. 217-222, 1995.

[8] S. Kleinstaubler and N. Sepeshri, "A polynomial network modeling approach to a class of large-scale hydraulic systems", *Computers Elect. Eng.* 22, pp. 151-168, 1996.

[9] S.-K. Oh and W. Pedrycz, "The design of self-organizing Polynomial Neural Networks", *Information Science*, Vol. 141, pp. 237-258, 2002.

[10] S.-K. Oh, W. Pedrycz and B.-J. Park, "Polynomial Neural Networks Architecture: Analysis and Design", *Computers and Electrical Engineering*, Vol. 29, Issue 6, pp. 703-725, 2003.

[11] Holland, J. H., *Adaptation In Natural and Artificial Systems*. The University of Michigan Press, Ann Arbour. 1975.

[12] D. E. Goldberg, *Genetic Algorithm in search, Optimization & Machine Learning*, Addison wesley, 1989.

[13] K. De Jong. Are genetic algorithms function optimizers? In *Proc. of PPSN II (Parallel Problem Solving from Nature)*, pages 3-13, Amsterdam, North Holland, 1992.

[14] Z. Michalewicz, *Genetic Algorithms + Data Structures = Evolution Programs*, Springer-Verlag, Berlin Heidelberg, 1996.

[15] S.-K. Oh and W. Pedrycz, "Identification of Fuzzy Systems by means of an Auto-Tuning Algorithm and Its Application to Nonlinear Systems", *Fuzzy sets and Systems*, Vol. 115, No. 2, pp. 205-230, 2000.

[16] D. E. Box and G. M. Jenkins, *Time Series Analysis, Forecasting and Control*, California: Holden Day, 1976.

[17] G. Vachtsevanos, V. Ramani, and T. W. Hwang, "Prediction of Gas Turbine NOx Emissions using Polynomial Neural Network", Technical Report, Georgia Institute of Technology, Atlanta, 1995.

[18] R. M. Tong, "The evaluation of fuzzy models derived from experimental data", *Fuzzy Sets and Systems*, Vol. 13, pp. 1-12, 1980.

[19] M. Sugeno and T. Yasukawa, "A Fuzzy-Logic-Based Approach to Qualitative Modeling", *IEEE Trans. Fuzzy Systems*, Vol. 1, No. 1, pp. 7-31, 1993.

- [20] C. W. Xu, and Y. Zailu, "Fuzzy model identification self-learning for dynamic system", IEEE Trans. on Syst. Man, Cybern., Vol. SMC-17, No.4, pp.683-689, 1987.
- [21] W. Pedrycz, "An identification algorithm in fuzzy relational system", Fuzzy Sets and Systems, Vol. 13, pp.153-167, 1984.
- [22] E. T. Kim, H. J. Lee, M. K. Park and M. Park, "A simple identified Sugeno-type fuzzy model via double clustering", Information Science 110, pp. 25-39, 1998.
- [23] Y. Lin, G. A. Cunningham III, "A new approach to fuzzy-neural modeling", IEEE Trans. Fuzzy Systems Vol. 3, No. 2, pp. 190-197, 1995.
- [24] T.-C. Ahn, S.-K. Oh, A thesis of emission pattern model about the atmosphere pollution material of a power plant, Electrical Engineering & Science Research Institute, Korea, 1997(in Korean).
- [25] S.-K. Oh, W. Pedrycz and H.-S. Park, "Hybrid Identification in Fuzzy-Neural Networks", Fuzzy Sets & Systems, Vol. 138, pp. 399-426, 2003.
- [26] S.-K. Oh, W. Pedrycz and H.-S. Park, "Rule-based Multi-FNN Identification with the Aid of Evolutionary Fuzzy Granulation", Journal of Knowledge-Based Systems, 2003(In press).
- [27] H.-S. Park, B.-J. Park and S.-K. Oh, "Optimal Design of Self-Organizing Polynomial Neural Networks By Means of Genetic Algorithms", Journal of the Research Institute of Engineering Technology Development, Vol. 22, pp. 111-121, 2002.
- [28] 오성권, "C 프로그래밍에 의한 퍼지모델 및 제어시스템", 내하출판사, 2002. 1.
- [29] 오성권, "프로그래밍에 의한 컴퓨터지능(퍼지, 신경회로망 및 유전자알고리즘을 중심으로)", 내하출판사, 2002. 8.

## 저 자 소 개



### 박 호 성 (朴 浩 成)

1999년 원광대 제어계측공학과 졸업. 동대학원 석사(2001), 2001~현재 동대학 박사과정. 관심분야는 퍼지, 뉴럴네트워크 및 뉴로-퍼지등의 지능형 모델링과 제어. 시스템 자동화 등.

Tel : (063) 850-6342

E-mail : neuron@wonkwang.ac.kr



### 박 병 준 (朴 炳 俊)

1998년 원광대 제어계측공학과 졸업. 동대학원 석사(2000). 동대학원 박사(2003). 관심분야는 시스템자동화, 퍼지이론 및 뉴로퍼지네트워크 응용, 지능형 모델링 및 계산지능 등.

Tel : (063) 850-6342

E-mail : lcap@wonkwang.ac.kr



### 장 성 환 (張 成 煥)

1974년 홍익대 전기공학과 졸업. 1982년 연세대 대학원 전기공학과 졸업(공학 석사, 박사). 1985-1986년 Northwestern 대학 전기 및 컴퓨터공학과 방문교수. 1980~현재 원광대 전기전자 및 정보공학부 교수. 관심분야는 자동제어, 지능제어 및 응용 등.

Tel : (063) 850-6734

E-mail : swhjang@wonkwang.ac.kr



### 오 성 권 (吳 聖 權)

1981년 연세대학교 전기공학과 졸업. 1983-89년 금성산전연구소(선임연구원). 1993년 연세대 대학원 전기공학과 졸업(공학박사). 1996-1997년 캐나다 Manitoba대학 전기 및 컴퓨터공학과 Post-Doc. 1993년~현재 원광대 전기전자 및 정보공학부 부교수. 2002년~현재 대한전기학회 및 제어·자동화·시스템공학회 편집위원, 관심분야는 시스템자동화, 퍼지이론 및 뉴로퍼지네트워크 응용, 계산지능 및 소프트 컴퓨팅 등.

Tel : (063) 850-6342

E-mail : ohsk@wonkwang.ac.kr